Stochastic Gradient Descent default hyperparameters:

'n0':1e-5,

'beta':1e-4,

'epsilon':1e-6,

'maxiters':10

NOTE: Average between training and validation must be weighted avg. (0.9\*training + 0.1\*valid)

Compare Closed Form and Stochastic Gradient Descent with default hyperparameters, with the 3 original features (No Text):

|  |  |  |
| --- | --- | --- |
| Metric | Closed Form | Gradient Descent |
| Time to run(s) | 0.2451 | 0.2303 |
| Training MSE | 1.085 | 1.1307 |
| Validation MSE | 1.0203 | 1.0734 |
| Average(V&T) MSE |  |  |

Using SGD with default hyperparameters, without scaling text features, or taking out stopwords, compare:

|  |  |  |
| --- | --- | --- |
| Metric | No Text | Basic 60 Text Features |
| Time to run(s) | 0.2303 | 2.8125 |
| Training MSE | 1.1307 | 68,223.7862 |
| Validation MSE | 1.0734 | 78,610.4992 |

Using SGD with default hyperparameters, with scaling text\_features so that they all fell within the 0 to 1 range, and multiplied that number by 6 to account for the overwhelming number of 0’s in the text\_features.

|  |  |  |  |
| --- | --- | --- | --- |
| Metric | No Text | Basic 160 Text Features | Basic 60 Text Features |
| Time to run(s) | 0.2303 | 7.4521 | 2.8125 |
| Training MSE | 1.1307 | 1.1289 | 1.1290 |
| Validation MSE | 1.0734 | 1.0681 | 1.0682 |

Using SGD with default hyperparameters, scaling factor above, and best above (160 text features), we now take out the stopwords.

|  |  |  |
| --- | --- | --- |
| Metric | No Stopwords | With Stopwords |
| Time to run(s) | 7.4521 | 6.7399 |
| Training MSE | 1.1289 | 1.1290 |
| Validation MSE | 1.0681 | 1.0681 |

Improvement isn’t drastic/ may not exist at all.

Using all the preprocessing metrics above, we will now change some important hyperparameters to see how performance of above model in SGD changes.

Using best above, find the best configuration below

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Metric | Alpha=  1e-8 | Alpha=1e-7 | Alpha\_0 = 1e-6 | Alpha\_0=1e-5 | Alpha\_0=1e-4 |
| Time to run(s) | 6.7869 | 6.7251 | 7.0901 | 6.3121 | 6.5704 |
| Training MSE | 2.1056 | 2.0462 | 1.6203 | 1.1290 | 1.88e9 |
| Validation MSE | 2.0660 | 2.0040 | 1.5611 | 1.0681 | 2.01e9 |

For the purposes of this assignment, we don’t have time to exhaustively search the hyper parameters. It is possible that when alpha=1e-4, it actually converges for different values of Beta, but due to lack of time and tools, and it being vastly past the scope of the assignment, we will not further search for the complete pareto-optimal configuration.

**Hyper\_Parameter Optimization**

We will now take alpha=1e-5, our default value, and move forward. Now we try different values of Beta.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Beta | 1e-2 | 1e-3 | 1e-4 | 1e-5 |
| Time to run(s) | 9.3341 | 9.1246 | 6.6821 | 7.4012 |
| Training MSE | 1.1293 | 1.1290 | 1.1290 | 1.1290 |
| Validation MSE | 1.0685 | 1.0682 | 1.0682 | 1.0681 |

Taking Beta=1e-5, as it starts to plateau here:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Iterations | 10 | 100 | 1000 | 10000 |
| Time to run(s) | 7.4012 | 7.1337 | 8.8842 | 30.544 |
| Training MSE | 1.1290 | 1.0906 | 1.0725 | 1.0569 |
| Validation MSE | 1.0681 | 1.0294 | 1.0184 | 1.1534 |

Now that we have our optimal max\_iterations: 100. After that, the data starts to overfit. From now on we will use 100 as our default hyperparameter for max\_iterations.

**New features**

All models considered here take the best parameter/preprocessing step taken from above unless otherwise specified.

Swear words:

We have text features

|  |  |  |  |
| --- | --- | --- | --- |
| Metric | Original features | Ofeatures + swear word count | Ofeatures + Length of comments |
| Time to run(s) | 7.1337 | 6.5415 | 7.0811 |
| Training MSE | 1.0906 | 1.0884 | Inf |
| Validation MSE | 1.0294 | 1.0237 | Inf |

It doesn’t look great for our features… the swear word count gives us an okay improvement, but still leaves some to be desired. while the length of the comments has our system completely diverging. Yet, we felt confident that these features could improve our model. After fidgeting, we discretized both of them.

|  |  |  |  |
| --- | --- | --- | --- |
| Metric | Binary swear words | Binary length | Both combined |
| Time to run(s) | 6.9025 | 7.2486 | 6.5823 |
| Training MSE | 1.0886 | 1.0888 | 1.0872 |
| Validation MSE | 1.0248 | 1.0209 | 1.0195 |

Now picking our best network. To review here is the spec:

Alpha = 1e-5, Beta = 1e-5, epsilon = 1e-9, iterations=100

Using standard text features with no stopword filtering, using 160 words, and using discrete swear\_words and discretized comment length feature:

|  |  |
| --- | --- |
| Metric | Best network run on test set |
| Time to run(s) | 6.1738 |
| Training MSE | 1.0873 |
| Validation MSE | 1.2947 |

Now using the same features, but with the closed form solution:

|  |  |
| --- | --- |
| Metric | Closed Form Solution |
| Time to run(s) | 7.0545 |
| Training MSE | 1.0462 |
| Validation MSE | 1.4650 |

So in the end, our best model still overfit. This is probably due to the distribution of the dataset, and if we used something like k fold cross validation, this answer would be more consistent with our training data.